Turkish Journal of Electrical Engineering & Computer Sciences

http://journals.tubitak.gov.tr/elektrik/

Research Article

Turk J Elec Eng & Comp Sci
(2013) 21: 851 – 863

doi:10.3906/elk-1112-71

An automated signal alignment algorithm based on dynamic time warping for capillary electrophoresis data

Fethullah KARABİBER*
Department of Chemistry, University of North Carolina, Chapel Hill, NC 27599-3290, USA

Received: 23.12.2011 • **Accepted:** 22.02.2012 • **Published Online:** 03.05.2013 • **Printed:** 27.05.2013

Abstract: Correcting the retention time variation and measuring the similarity of time series is one of the most popular challenges in the area of analyzing capillary electrophoresis (CE) data. In this study, an automated signal alignment method is proposed by modifying the dynamic time warping (DTW) approach to align the time-series data. Preprocessing tools and further optimizations were developed to increase the performance of the algorithm. As a demonstrative case study, the developed algorithm is applied to the analysis of CE data from a selective 2'-hydroxyl acylation analyzed by primer extension (SHAPE) evaluation of the RNA secondary structure. The time-shift problem is one of the main components in the analysis of the SHAPE data. The accuracy and execution time of the algorithm are illustrated with experimental results obtained by applying to different types of data. The experimental results show that the signal alignment algorithm efficiently corrects the retention time variation. The developed tools can be readily adapted for the analysis of other biological datasets or time series.

Key words: Bioinformatics, time series, signal alignment, dynamic time warping, capillary electrophoresis

1. Introduction

Bioinformatics is the application of computer sciences and mathematics to the management and analysis of complex datasets to aid the solution of biological problems [1]. The alignment of time-scaled and time-shifted signals is often necessary in the analysis of datasets obtained from biological experiments [2]. Time shifts can occur when a signal is measured as a function of time for 2 or more datasets with small- or large-scale differences in the experimental conditions across repeated samples. The differences could be due to some factors including temperature or voltage changes, instrument imperfections, or variations in the flow rates. Thus, the comparison of different samples can be complicated by differences in their time scales or differences in the lengths of the sample vectors.

In order to correct the retention time drift, many different approaches were proposed in the literature. One of the well-known algorithms to compare 2 discrete signals or time series is dynamic time warping (DTW) [3]. DTW is based on dynamic programming, which is a method of solving complex problems by breaking them down into simpler steps [4]. DTW is a fast and efficient method for the alignment of time-dependent sequences. Although it was originally developed for speech recognition [3], the classical DTW and its different variations have also been applied to many other fields, such as signature similarity [5], clustering [6], data mining and information retrieval [7], computer vision [8], and chemical engineering [9]. The details of DTW and

*Correspondence: fkarabiber@unc.edu

references to its use in different areas can be found in [10–12]. In addition to DTW, some other approaches were proposed to correct the retention time drift. Gong et al. [13] used the combination of chemometric resolution and cubic spline data interpolation to correct the retention time shifts for the chromatographic fingerprints of herbal medicines. Correlation optimized warping was used to align chromatographic data in [14]. An approach to align gas chromatography–mass spectrometry was proposed based on dynamic programming and peak similarity in [15].

The widely used selective 2'-hydroxyl acylation analyzed by primer extension (SHAPE) [16] and hydroxyl radical nucleic acid probing experiments present important examples of the time-shift problem. Since the experimental parameters that characterize each capillary may differ, the measured traces across the capillaries can vary in time, velocity, and intensity. As can be seen from Figure 1, the signals obtained from different experiments are similar but out of phase. Since each reaction is analyzed using a DNA primer labeled with a different fluorophore, the dyes alter the electrophoretic migration rates so that traces in the same capillary have slightly different elution times. The ShapeFinder software [17] was developed and has been widely used to analyze the SHAPE data. An important part of the analysis of the raw SHAPE capillary electrophoresis (CE) data is to align all of the traces in the same time scale. ShapeFinder uses several mobility shift tools to correct the time offsets by initiating the parameters manually. This process is time-consuming to implement (about 15–30 min) and must be optimized for each dataset.

In this work, a new automated signal alignment algorithm is developed by modifying DTW to solve the shift problems for SHAPE and other nucleic acid chemical probing data obtained by CE. The algorithm infers whether 2 time series are homologous by calculating the similarity score between them. The algorithm optimally aligns the 2 time series to maximize the similarity. In addition, preprocessing tools and further optimizations were developed to increase the performance of the algorithm.

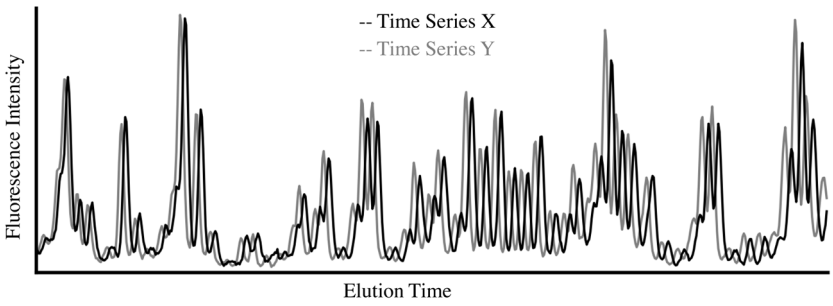

Figure 1. The raw data (sequencing ladders) obtained from 2 different capillaries. Time series X (black) and time series Y (gray) are similar but out of phase.

2. Materials and methods

2.1. SHAPE data

SHAPE measures the local backbone flexibility at nearly every position in an RNA. The analysis of a SHAPE experiment is important to extract the quantitative, single nucleotide resolution reactivity information. The output of a SHAPE experiment resolved by CE is an electropherogram or trace. An electropherogram contains 3 or 4 individual channels that report the fluorescence intensity versus the elution time information, where each channel corresponds roughly to 1 of the SHAPE reactions. In a SHAPE experiment, there are 3 or 4 individual channels: plus SHAPE reagent (RX), without SHAPE reagent or background (BG), and 1 or 2 sequencing

ladders (SLs). The RX, BG, and SLs are labeled with a different fluorophore, and these are then mixed and loaded onto a single capillary. Unprocessed electropherogram data are exported from ABIF type (.fsa) files that are generated by CE (Applied Biosystems AB3130 instrument) [16,17].

2.2. Preprocessing

The appropriate and essential preprocessing tools improve the performance of the signal alignment algorithm significantly. The implemented tools are given below sequentially.

Smoothing: The raw data exported from the experiments have many kinds of noise, such as high and low frequency, which are the rapid changes in the amplitude from point to point within the signal. The triangular smooth method [18], which is one of the most common smoothing methods, is used to reduce the noises. The triangular smooth is like the rectangular smooth, except that it implements a weighted smoothing function [18]. The smooth coefficients are symmetrically balanced around the central point. Since in a SHAPE experiment the peak is the most important of the measurement objectives, it is important to preserve the peaks and other features in the signal. Triangular smooth not only reduces the noise but also preserves the peak shapes.

Signal enhancement: After smoothing, some peaks may be distorted. To obtain the peaks more accurately, a second derivative-based resolution enhancement technique may be applied. In the enhancement method, the second derivative of the input signal is subtracted from the input. The useful feature of this procedure is that it does not change the total peak area because the total area under the curve of the derivative of a peak-shaped signal is zero [18].

Baseline adjustment: Fluorescent background noise causes the baseline in each channel to drift. The baseline adjustment algorithm is used to remove the background signal and to normalize the baseline. In this algorithm, the minimum signal intensity points within the specified window size are found and then the baseline signal is obtained by applying the linear interpolation using the minima points. Finally, the obtained baseline signal is subtracted from the input signal to adjust the baseline [17].

Normalization: Since the concentrations of the chemicals used in the experiments may vary or because the detection equipment is imperfect, the experimentally derived data commonly have experimental biases. Normalization is usually used to remove such biases to compare the experiments. In this study, the commonly employed zero-mean, unit-variance statistical normalization is used. The mean of the data is subtracted from each data point and then these differences are divided by the standard deviation of the data to obtain normalized data [19].

2.3. The theory of classical DTW

The DTW algorithm finds an optimal match between 2 time series. The 2 time series' data are nonlinearly warped in such a way that the similar regions are aligned and a minimum distance between them is obtained. DTW works by warping the time axis iteratively until an optimal match between the 2 sequences is found. Here, the summary of the theory of classical DTW is given. The details and pseudocodes of the classical DTW can be found in [10,12]. Classical DTW can be implemented in 3 main steps.

Step 1. Suppose that we have 2 time series $X = (x_1, x_2, .., x_i, .., x_N)$ and $Y = (y_1, y_2, .., y_j, .., y_M)$, of length N and M, respectively. An N-by-M matrix C called the distance or local cost is created to represent the distance of each pair of elements of the time series X and Y. In the distance matrix, the $(i\,\text{th}, j\,\text{th})$ element of the matrix is the distance between the 2 points x_i and y_j, and is computed by a function called the distance or cost function. The most common function used to compute the distance is the Euclidean distance function,

as shown in Eq. (1).

$$C(i,j) = d(x_i, y_j) = (x_i - y_j)^2 \tag{1}$$

Step 2. Using the cost matrix, the accumulated cost matrix D is defined as follows. The first row and first column of matrix D are calculated and then all of the other elements are computed using the following statement:

$$\begin{aligned} &D(i,1) = \sum_{k=1}^{i} C(k,1), \text{ for } i \in [1,N] \\ &D(1,j) = \sum_{k=1}^{j} C(1,k), \text{ for } j \in [1,M] \\ &D(i,j) = C(i,j) + min\{D(i-1,j-1), D(i-1,j), D(i,j-1)\}, \text{ for } i \in [2,N] and j \in [2,M] \end{aligned} \quad . \tag{2}$$

Step 3. After obtaining the accumulated cost matrix, the optimal alignment path is found. An optimal warping path between X and Y is a warping path having a minimal cost among all of the possible warping paths. A warping path, W, is a contiguous set of matrix elements that assigns the elements of X and Y to each other. $w_k = (i,j)_k$ is defined as the kth element of W and

$$W = (w_1, w_2, .., w_k, ..w_K), max(N,M) \leq K < N + M + 1. \tag{3}$$

The warping path is typically subjected to several conditions and is obtained by taking into consideration the following conditions.

- **Boundary:** The first and last elements of X and Y are matched to each other. The warping path starts and finishes in the diagonally opposite corner of the accumulated cost matrix D, namely, $w_1 =$ *(1, 1)* and $w_K = (N,M)$.

- **Continuity or step size:** This condition restricts the allowable steps in the warping path to adjacent cells. No element in X and Y can be omitted and there are no replications in the alignment. Given that $w_k = (i,j)$, then $w_{k-1} =$ *(i', j')*, where $i - i' \leq$ *1* and $j - j' \leq$ *1*.

- **Monotonicity:** This limits the warping path from long jumps while aligning the sequences. The following condition forces the points in W to be monotonically spaced in time. Given that $w_k = (i,j)$, then $w_{k-1} =$ *(i', j')*, where $i - i' >$ *0* and $j - j' >$ *0*.

The optimal warping path is calculated by satisfying the constraints given above with minimal cost. The warping path could be found by simple backtracking from $w_K = (N,M)$ to $w_1 =$ *(1,1)*, using the following statements:

$$w_{k-1} = \begin{cases} (1, j-1), & \text{if } i = 1 \\ (i-1, 1), & \text{if } j = 1 \\ argmin\{D(i-1,j-1), D(i-1,j), D(i,j-1)\}, & otherwise \end{cases} \quad . \tag{4}$$

A sample accumulated cost matrix and optimal warping path is illustrated in Figure 2a. Using the warping path shown as a white line, the signals can be matched as shown in Figure 2b.

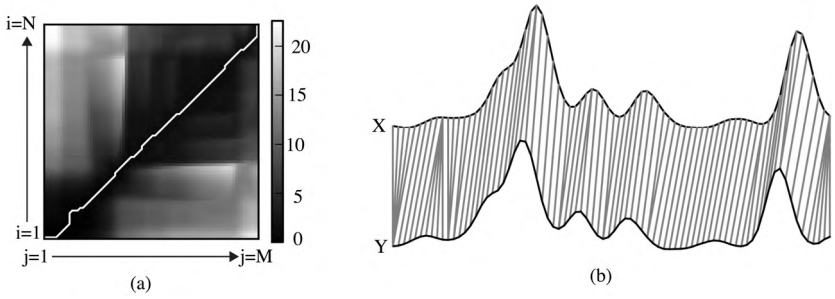

Figure 2. Representation of the classical DTW. a) A warping matrix is constructed and searched for the optimal warping path to align the signals. The color map represents the value of the accumulated cost matrix. The optimal warping path is drawn with a white line. b) The result of the alignment using the optimal warping path. Lines represent the matched data points.

2.4. Subsequence DTW

In many applications, the signals to be compared may have a significant difference in length. Instead of aligning these sequences globally, a subsequence within a longer signal is found to fit the shorter signal optimally. A local signal alignment algorithm is used to identify the segment within the longer signal that is the most similar to the shorter one. The problem of finding the optimal subsequence can be solved by adapting the classical DTW. Details about the local alignment can be found in [10,20].

Let X and Y be 2 signals, where we assume that the length of X is larger than the length of Y. An optimal alignment between X and Y can be computed by some modifications in the classical DTW algorithm described in Section 2.3. The first modification is made in the initialization (Step 2) of the DTW algorithm. The basic idea is to not penalize the omissions in the alignment between X and Y that appear at the beginning and at the end of Y. More precisely, the accumulated cost matrix for the first column is defined as:

$$D(i,1) = C(i,1) for i \in [1,N]. \tag{5}$$

The second modification is done when finding the optimal warping path. Instead of starting in the top-right corner of the D matrix, the path starts at a location with the minimum value in the last column of D. The warping continues until all of the data points in Y are matched with a point in X. In other words, the warping path can be found by simple backtracking from the points $w_K = \textit{(argmin (D(1:N, M)), M)}$ to the $j = 1$ or $i = 1$.

The sample accumulated cost matrix D and warping part with the alignment result are given in Figure 3, where it can be seen that the shorter signal is aligned correctly with a subsequence within the longer one.

2.5. Implemented DTW modifications

Various modifications have been proposed to speed up DTW computations as well as to control the routes of the warping path in a better way [12]. One of the common DTW variants is to impose global constraint conditions on an admissible warping path. Two well-known global constraint regions are the Sakoe–Chiba band and the Itakura parallelogram [3,12]. In this application, the Sakoe–Chiba band, which runs along the main diagonal and has a fixed (horizontal and vertical) width $T \in N$, is used as a global constraint. The alignment of the time points can be selected only from the defined region. Moreover, the distance function is not calculated in all of the data points, but only in a defined region. This yields a lower execution time.

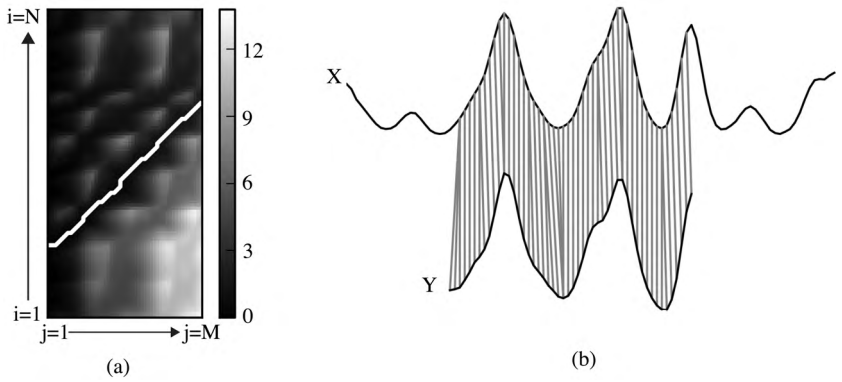

Figure 3. The local signal alignment by modified DTW. a) An accumulated cost matrix with an optimal warping path. Notice that the start and end of the optimal warping path is shown by a white line. b) Result of the local signal alignment.

The derivative DTW proposed in [11] does not calculate the distances between the intensity of the data points, but calculates them between their associated first-order derivatives. Since synchronization is based on the shape characteristics (slopes, peaks) rather than the values, the derivative DTW gives better results than the classical DTW, especially for the signals that have peaks and baseline drifts. The derivative of a data point is given in the following equation:

$$f'(x_i) = ((x_i - x_{i-1}) - (x_{i+1} - x_{i-1})/2)/2, for 1 < i < N. \tag{6}$$

Since the distance function is used to determine the similarity between the elements of the 2 sequences, the choice of a suitable local cost or distance function is of crucial importance. In the classical DTW, the Euclidean distance function is commonly used for the similarity of time points to obtain the cost matrix [10]. Since the Euclidean distance function computes the distance just using the power of the difference (Eq. (1)), it is not enough to obtain the optimum cost matrix for the SHAPE data. In order to obtain a better warping path, a new distance function is proposed. The formula for the newly proposed distance function is:

$$d(x_i, y_j) = |x_i - y_j| \times \exp(|i - j| \times T) \times P. \tag{7}$$

Here, T is the elution time tolerance parameter, which determines the importance of the elution time to the distance score. In other words, T determines the growing rate of the exponential function. The T value may be between 0 and 1. If $T = 0$, there is no effect of the time on the distance. The time difference penalty will be more effective for a higher value of T. For the examples shown here, T was 0.05.

In order to calculate the reactivity of RNA using a SHAPE experiment, peaks in the traces should be aligned correctly. Since the peaks are the most important features of the traces, giving priority to the peak positions yields better results. If both x_i and y_j are the peak positions, the distance of these points is decreased by a factor of 20% (for P = 0.8). Otherwise, P is assigned to 1. Hence, the peak points will have advantages if they are aligned. Since the derivatives of the signals are used in the proposed DTW, the peak positions can be found easily by finding the zero-crossing points in the derivative.

At this point, the algorithm for the new distance function is given in *Algorithm-1* as a pseudocode. Note that notations used for the pseudocode in this paper are given as the following. Variables, such as a or *total*, contain some scalar values. An array of N elements is denoted like *X[1,2,...,N]*. $X[i]$ represents the ith element

of array X and $C[i,j]$ represents the ith row and jth column of the $N \times M$ matrix C. General functions, such as the minimum and absolute value, are written in as capital letters with parentheses. For example, *MAX(X)* would return the maximum elements of array X. The arrow ($\leftarrow$) represents the assignment. Loops and conditional statements are written in lowercase and bold letters. Comments that begin with "//" give information about the code line.

```
Algorithm-1:  New Distance Matrix
INPUT: Derivatives of two time series (X', Y') and time tolerance (T)
OUTPUT: Cost matrix C
1:  for i ← 1 to N do
2:    for j ← 1 to M do
3:       C[i,j]= ABS(X' [i] - Y' [j]) x EXP(ABS(i − j)*T)
         // Control the points for the peak using downward zero crossing
4:       if SIGN (X' [i]) == -1 and SIGN (X' [i-1]) == 1 then
5:          if SIGN (Y' [j]) == -1 and SIGN(Y' [j-1]) == 1 then
6:            C[i, j] ← C[i, j] x 0.8
```

2.6. Peak matching using the warping path

The next step is to determine which of the peaks found in different samples have a common origin. *Algorithm-2* is used to match a peak in the 1st trace to a peak in the 2nd trace. After finding the optimal warping path, the peak detection is applied to the data. Peak detection is performed by looking for the downward zero-crossing in the first derivative of the time-series datasets [18]. If the sign of the derivative changes from positive to negative, it means that there is a peak at this point. After the peak detection, the peak positions are matched using the warping path. However, a correlation is used to verify whether these 2 peaks are identical. If the peak positions in the warping path are matched and the correlation result is above 95%, these peak positions will be used for the next step. At the end of this process, a matrix containing the peak matching points is obtained. For further optimizations, the widths of the matched peaks are controlled.

```
Algorithm-2:  Find the peak match points using warping path
INPUT: Signals (X[1,2,..N]) and Y[1,2,...,M]), warping path (WP[[1,2];[1,2,...,K]]
OUTPUT: Matched peaks (MPX,MPY)
// Obtain arrays for the peak positions in X and Y signals.
1:  PX← PEAKDETECTION(X), PY← PEAKDETECTION(Y)
2:  for i ← 1 to LENGTH(PX) do
3:  j ← WP[2,PX[i]]
    // Control whether there is a peak in pY corresponding the peak in pX.
4:  if (j in PY) == True then
     // Use correlation to make sure whether these peaks are similar
5:   R ← CORRELATION(X[PX[i]-5 :  PX[i]+5], Y[PY[j]-5 :  PY[j]+5] )
6:   if R >= 0.95 then
7:     APPEND(MPX,i) ; APPEND(MPY,j)// Append values to the end of an array.
```

2.7. Signal stretch and compress

After obtaining the peak matching points, cubic spline interpolation is used to compress or stretch the signal to align the signals. In numerical analysis, cubic splines are often used in practice because of the simplicity of their construction. They produce a curve that appears to be seamless and they avoid oscillation problems in the curve fit [19]. The objective is to fit a cubic spline for the data points. A typical curve fit involves forming one equation through all of the points. A spline allows each segment to have a unique equation constraining the curve fit to the data properties. The cubic spline method avoids oscillation problems in the curve fit connecting the individual segments. In general, the cubic spline provides a good curve fit for the arbitrary data points [19]. A signal stretch and compress algorithm based on the cubic spline is performed using *Algorithm-3* to obtain the aligned signals.

```
Algorithm-3:  Signal Stretch and Compress
INPUT: The signals X and Y, corresponding matched peak arrays (MPX, MPY)
OUTPUT: Aligned signal (newY[1,2,...,N] )
1:  for i←1 to LENGTH(MPY) do
       // Calculate length of the consecutive match points in MX and MY
2:     d1← MPX[i+1] - MPX[i] ; d2← MPY[i+1] - MPY[i]
3:     if d1==d2 then
4:          newY [MPX[i] :  MPX[i+1]] = Y [MPY[i] :  MPY[i+1]]
5:     else
         // Find the Cubic Spline representation of signal part
6:       Coefficient=INTERPOLATE (Y[MPY[i] :  MPY[i+1]])
         // Create new array of size d1 between MPY[i] and MPY[i+1]
7:       array = LINSPACE (MPY[i], MPY[i+1], d1)
         // Calculate new values for newY using array
8:       newY[MPX[i] :  MPX[i+1]] = Coefficient(array)
```

2.8. Proposed signal alignment algorithm

In order to improve the signal alignment, some prepressing tools, various modifications for DTW, and further processes such as *Algorithm-2* and *Algorithm-3* are employed. The details of the proposed signal alignment algorithm, which includes 6 main steps, are given in *Algorithm-4*.

3. Results and discussion

3.1. Aligning signals across capillaries

One of the main challenges with the electrophoresis data is to align the data obtained from different experiments. Since the parameters used in each experiment, such as the temperature and voltage, are different, data may vary in time, speed, or intensity. In a SHAPE experiment, different capillaries may have different elution times and intensities. To overcome this problem, signals should be aligned in terms of the elution time. Since the sequencing traces are similar, these traces are used to align signals across the capillaries. As can be seen in Figure 4a, the pattern of the signals is similar, but the elution time and intensity are different from each other.

Algorithm-4: Proposed signal alignment algorithm
INPUT: Raw data X[1,2,..N] and Y[1,2,...,M]
OUTPUT: Aligned signal (newY[1,2,...,N])
Step 1 -- Apply preprocessing tools to smooth, enhance, adjust baseline, and normalize the input signals.
Step 2 -- Compute derivative of the preprocessed signals and obtain cost matrix using *Algorithm-1*.
Step 3 -- Obtain accumulated cost matrix using Sakoe--Chiba band, taking into account of the global constraints
Step 4 -- Obtain optimal warping path from accumulated cost matrix using defined conditions and the statement given in Step 3 of the theory of classical DTW.
Step 5 -- Obtain peak match using *Algorithm-2*. Control the width of the consecutive matched peaks.
Step 6 -- Compress and stretch Y using *Algorithm-3* using final matched peaks to get newY.

In order to align the sequencing traces, a DTW-based signal alignment algorithm (*Algorithm-4*) is used. In order to show the accuracy of the signal alignment algorithm, 4 different sequencing data from 4 different capillaries are employed. As can be seen in Figure 4a, the signals have a time-shift problem. After applying the proposed signal alignment algorithm, all of the time-series data are aligned perfectly, as given in Figure 4b.

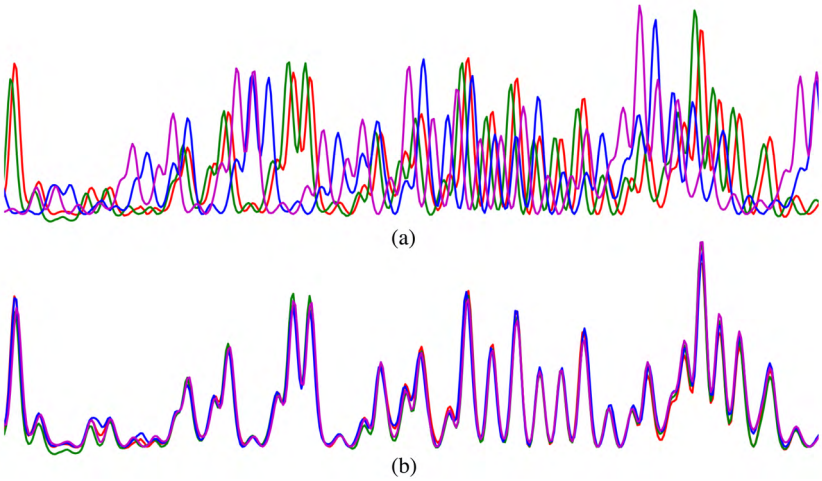

Figure 4. Signal alignment across capillaries: a) 4 different datasets have time-shift problems; b) result of the proposed signal alignment algorithm. A signal is selected as a reference and the other sample signals are aligned to the reference signal.

3.2. Aligning signals within a capillary

In any separation involving multiple dyes linked to DNA, the different dye molecules will alter the relative speed at which the attached DNA fragments travel through the capillary column. This is primarily caused by the differing molecular weights of the fluorescent dyes. As a result, the data corresponding to the same DNA lengths elute at slightly different times [16,17]. In order to align the signals in the capillary, the properties of the dyes are used. Each dye has a different wavelength and migration time. In a SHAPE experiment, the most commonly used dyes are VIC, NED, FAM, and JOE. As JOE and FAM have almost the same migration time,

VIC and NED have almost the same migration time. However, JOE and FAM are faster than VIC and NED. For example, supposing that FAM is used for the reagent and VIC is used for the sequence, the sequence signal should be shifted left.

As can be seen from Figure 5a, the amplitude of each peak and the pattern of the signals are completely different. However, in a SHAPE experiment, the lowest 20% of the peaks in the RX, BG, and SL signals are similar to each other. Step-5 in *Algorithm-4* is modified. Instead of all of the peaks, the lower peaks are matched. The algorithm is also improved using a shift direction determined by the dye types used for each reaction. In Figure 5a, a capillary with a RX and SLs is shown. It can be clearly seen that the signals are not aligned because of different dyes (VIC for reagent, NED for sequencing). After applying the shift correction algorithm defined above, the signals in the same capillary are successfully aligned, as shown in Figure 5b.

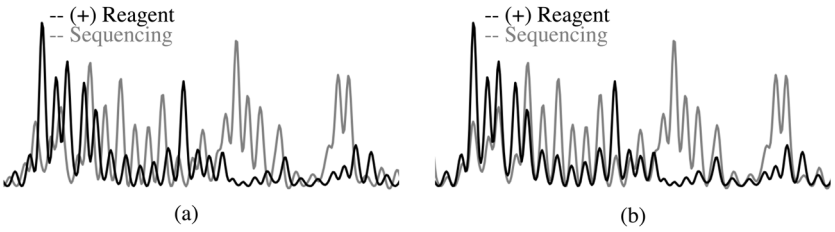

Figure 5. Signal alignment within a capillary: a) the signals in the same capillary with reagent (red) and sequence signal (green) have different time scale; b) result of the proposed signal alignment.

3.3. Defining region of interest automatically

The other application developed for the SHAPE data analysis is to define a region of interest automatically using the local signal alignment approach. In fluorescent primer-based sequencing, the sequence data is collected after the primer peak, which is the first wide peak in a trace (see Figure 6). In order to increase the accuracy of the next steps, it is useful to remove the data that do not contain any sequence information [17]. In order to find the same region of interest in traces from different capillaries, the local signal alignment algorithm may be used.

Since each capillary has the same ddNTP-SL, sequencing traces may be used to select the same region. After selecting the region of interest in the reference capillary manually, the selected data are used to find the same region in another capillary. The local peak alignment algorithm is applied using the reference and sample data. The first and last elements of the warping path are used to define the region of interest in the sample data.

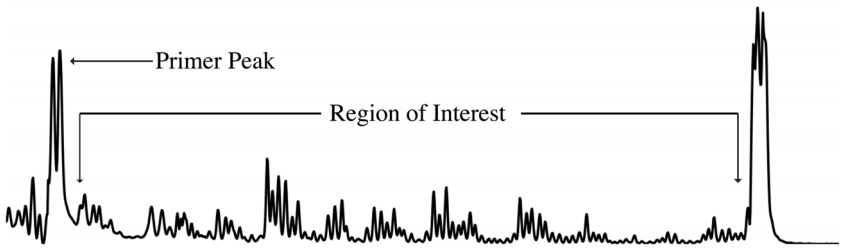

Figure 6. Representation of a region of interest and identification of a primer peak in a raw trace.

3.4. Comparison and discussion

The main advantages of the proposed algorithms are accuracy and speed. A comparison of the methods in terms of the execution time and accuracy is given in the Table. The implemented preprocessing tools increase the success of the DTW algorithm. In addition to the new distance function, the peak matching algorithm (*Algorithm-2*) and controlling of the width of the matched peaks reduce the effects of misaligned data points. The proposed procedure for signal alignment can correct the time-shift problem in an efficient and accurate way. The developed applications in Sections 3.1 and 3.2 solve the time-shift problems in the analysis of the SHAPE data analysis challenges successfully.

Table. Comparison of the methods in terms of the execution time for N = M =1000 and the accuracy for over 10,000 data points.

Method	Execution time	Accuracy
Classical DTW	0.0772 s	87.48%
Modified DTW	0.188 s	91.79%
ShapeFinder	5–10 min	N/A

In order to show the accuracy, the developed algorithms were tested on different time-series datasets. In order to compare the performance of the classical and modified DTW approaches in terms of accuracy, over 10,000 data points from different SHAPE experiments were evaluated. After applying the alignment algorithms the number of correctly matched data points was calculated to determine the accuracy. The classical DTW approach was able to achieve only 87.48% accuracy, whereas the proposed approach was 91.79% accurate To show the accuracy of the warping algorithms, the experimental results of the classical and modified DTW are given in Figures 7a and 7b, respectively.

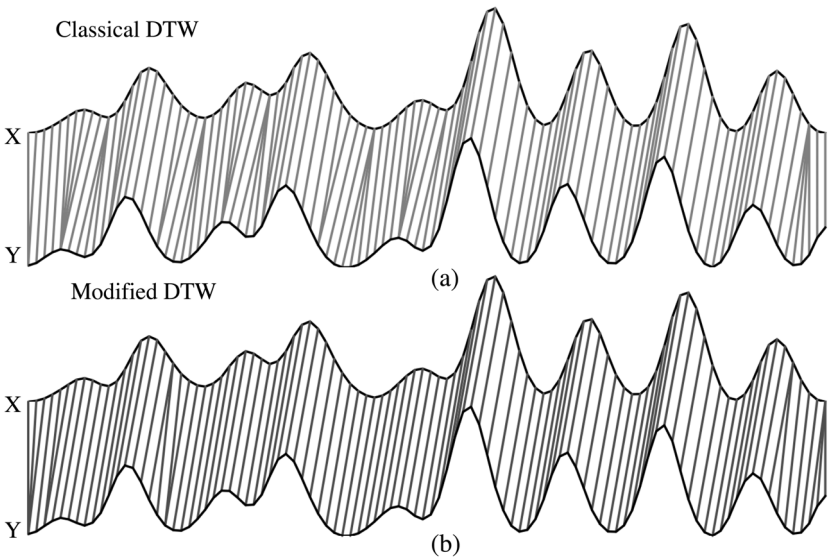

Figure 7. Comparison of classical and modified DTW. a) As can be clearly seen, there are many mismatches in the results of classical DTW. b) The alignment of 2 points in the data is performed more accurately by modified DTW.

In addition, the execution time of the proposed algorithm is incomparable with the algorithms used in ShapeFinder. Since the signal alignment is performed manually in ShapeFinder, the process can be done

carefully in 5–10 min, depending on the length of the traces. The execution time of the developed signal alignment algorithm for 1000 data points is below 1 s. The classical DTW is slightly faster than the proposed alignment algorithm due to different distance functions. In practical terms, this difference in the execution time is insignificant, but the accuracy of the proposed signal alignment algorithm is clearly superior. Note that the time space complexity of DTW is *O(NM)* in either case. The big O notation is used to show how an algorithm responds to changes in the input size in terms of the processing time or working space requirements.

4. Implementation

All of the developed methods are implemented using the Python programming language, version 2.6 [21]. The *NumPy* [22] and *SciPy* [22] packages, which are not part of the standard Python installation, are used to manipulate the array and data. *NumPy* is the fundamental package needed for scientific computing with Python. The *NumPy* package contains the array manipulation routines and the *SciPy* package contains a variety of scientific packages. In this study, the correlation, standard deviation, and other array manipulation routines are taken from *NumPy* and the Cubic Spline is taken from *SciPy*. *Matplotlib* [23] is used to draw the figures. *Matplotlib* is a Python 2D plotting library that produces quality figures and interactive environments across platforms. All of the packages are open-source software and can be downloaded from their website for free.

5. Conclusion

The data generated by CE of nucleic acid fragments can be corrected automatically for shifts in the elution time using the modified DTW approach. The main advantages of the developed algorithms for the signal alignment are speed and accuracy. Preprocessing tools and further optimizations were also developed to increase the performance of the algorithms. The test results prove that the time-shift problems, which are one of the most important components of the SHAPE data analysis, are solved correctly using the developed algorithms in a much shorter time. In addition, another challenge in the SHAPE data analysis is solved using the local signal alignment. These results encourage developing fully automated software to analyze the SHAPE data. As a result, a new distance function and other algorithmic features make possible the rapid signal alignment and highly accurate comparisons of complex time-series datasets.

Acknowledgments

I especially thank Dr Oleg Favorov and Dr Kevin Weeks for their feedback and suggestions. I also thank the members of the Weeks Laboratory at the University of North Carolina at Chapel Hill for providing the data. This work was supported by a grant from the US National Institutes of Health (AI068462).

References

[1] J. Kinser, Python for Bioinformatics, Burlington, Massachusetts, Jones and Bartlett Publishers, 2008.

[2] M. Last, A. Kandel, H. Bunke, Data Mining in Time Series Databases, Singapore, World Scientific, 2004.

[3] H. Sakoe, S. Chiba, "Dynamic programming algorithm optimization for spoken word recognition", IEEE Transactions on Acoustics, Speech and Signal Processing, Vol. 26, pp. 43–49, 1978.

[4] S.R. Eddy, "What is dynamic programming?", Nature Biotechnology, Vol. 22, pp. 909–910, 2004.

[5] M.E. Munich, P. Perona, "Visual identi?cation by signature tracking", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 25, pp. 200–217, 2003.

[6] V. Niennattrakul, C.A. Ratanamahatana, "On clustering multimedia time series data using K-means and dynamic time warping", International Conference on Multimedia and Ubiquitous Engineering, pp. 733–738, 2007.

[7] T. Kahveci, A. Singh, A. Gurel, "Similarity searching for multi-attribute sequences", Proceedings of the Scienti?c and Statistical Database Management, pp. 175–184, 2002.

[8] Z. Zhang, K. Huang, T. Tan, "Comparison of similarity measures for trajectory clustering in outdoor surveillance scenes", Proceedings of the 18th International Conference on Pattern Recognition, Vol. 3, pp. 1135–1138, 2006.

[9] J. Vial, H. Nocairi, P. Sassiat, S. Mallipatu, G. Cognon, D. Thiebaut, B. Teillet, D. Rutledge, "Combination of dynamic time warping and multivariate analysis for the comparison of comprehensive two-dimensional gas chromatograms application to plant extracts," Journal of Chromatography A, Vol. 1216, pp. 2866–2872, 2009.

[10] M. Müller, Information Retrieval for Music and Motion, Berlin, Springer, pp. 69–84, 2007.

[11] E.J. Keogh, M.J. Pazzani, "Derivative dynamic time warping", First SIAM International Conference on Data Mining, 2001.

[12] P. Senin, "Dynamic time warping algorithm review", Information and Computer Science Department, University of Hawaii, pp. 1–23, 2008.

[13] F. Gong, Y.Z. Liang, Y.S. Fung, F.T. Chau, "Correction of retention time shifts for chromatographic fingerprints of herbal medicines", Journal of Chromatography A, Vol. 1029, pp. 173–83, 2004.

[14] G. Tomasi, F. Van Den Berg, C. Andersson, "Correlation optimized warping and dynamic time warping as preprocessing methods for chromatographic data", Journal of Chemometrics, Vol. 18, pp. 231–241, 2004.

[15] M.D. Robinson, D.P. De Souza, W.W. Keen, E.C. Saunders, M.J. McConville, T.P. Speed, V.A. Likić, "A dynamic programming approach for the alignment of signal peaks in multiple gas chromatography-mass spectrometry experiments", BMC Bioinformatics, Vol. 8, p. 419, 2007.

[16] K.A. Wilkinson, E.J. Merino, K.M. Weeks, "Selective 2'-hydroxyl acylation analyzed by primer extension (SHAPE): quantitative RNA structure analysis at single nucleotide resolution", Nature Protocols, Vol. 1, pp. 1610–1616, 2006.

[17] S.M. Vasa, N. Guex, K.A. Wilkinson, K.M. Weeks, M.C. Giddings, "ShapeFinder: A software system for high-throughput quantitative analysis of nucleic acid reactivity information resolved by capillary electrophoresis", RNA, Vol. 14, pp. 1979–1990, 2008.

[18] T. O'Haver, "An introduction to signal processing in chemical analysis", available at http://terpconnect.umd.edu/∼toh/spectrum/, 2009.

[19] J. Kiusalaas, Numerical Methods in Engineering with Python, Cambridge, Cambridge University Press, 2010.

[20] T. Aruk, D. Ustek, O. Kursun, "A novel partial sequence alignment tool for finding large deletions", The Scientific World Journal, doi 10.1100/2012/694813, 2012.

[21] G. vanRossum, F.L. Drake (eds.), Python Reference Manual, available at http://www.python.org/, 2001.

[22] F. Jones, T. Oliphant, P. Peterson, "SciPy: open source scientific tools for Python", available at http://www.scipy.org/, 2001.

[23] J.D. Hunter, "Matplotlib: A 2D Graphics Environment", Computing in Science and Engineering, Vol. 9, pp. 90–95, 2007.